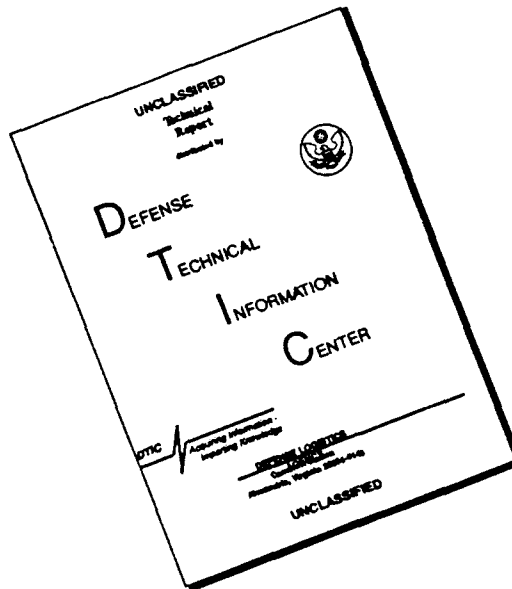


REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
<small>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.</small>				
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE May 1, 1994	3. REPORT TYPE AND DATES COVERED Final, September 7, 1993-February 7, 1994		
4. TITLE AND SUBTITLE Studies of Wave Propagation in Nonlinear Composite Materials		5. FUNDING NUMBERS G:N00014-93-1-6033		
6. AUTHOR(S) Chiu T. Law				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Johns Hopkins University Homewood Research Administration 105 Ames Hall, 3400 N. Charles Street Baltimore, MD 21218-2686		8. PERFORMING ORGANIZATION REPORT NUMBER		
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Contracting Officer Naval Research Laboratory 4555 Overlook Ave., S.W. Washington, DC 20375-5000		10. SPONSORING / MONITORING AGENCY REPORT NUMBER		
11. SUPPLEMENTARY NOTES				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Unlimited Availability		12b. DISTRIBUTION CODE		
13. ABSTRACT (Maximum 200 words) This project involves the investigation of the wave propagation in nonlinear composite media. This effort demonstrated theoretically that we can form a composite medium from a nonlinear medium with optical vortex soliton (OVS). The light-induced waveguide formed by the OVS can be a means for building an all-optical modulator. The dynamics of polarized OVS is also examined and remarkable evolution of instability emerges.				
14. SUBJECT TERMS nonlinear wave propagation, all-optical modulator, optical vortex soliton		15. NUMBER OF PAGES 22		
		16. PRICE CODE		
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT	

19960815 149-

**Personally Identifiable
Information Redacted**

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

Final Technical Report
Grant No.: N00014-93-1-G033

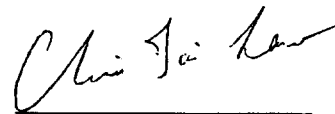
Studies of Wave Propagation in Nonlinear Composite Materials

Department of Electrical and Computer Engineering
Johns Hopkins University
Baltimore, MD 21218

Submitted to

Department of the Navy, Naval Research Laboratory
Scientific Officer: Dr. Anthony Campillo

by
Principal Investigator:



Dr. Chiu T. Law



Project Period: September 7, 1993 - February 7, 1994

May 1, 1994

1. Summary

This project involves the investigation of the wave propagation in nonlinear composite media. Works were mainly concentrated on obtaining analytical and numerical solutions of the (2+1)-D nonlinear Schrödinger equation, which accurately models nonlinear wave propagation. This effort demonstrated theoretically that we can form a composite medium from a nonlinear medium with optical vortex solitons (OVS). These spatial solitons actually induce ideal waveguides within the nonlinear medium. What is more, the characteristics of these waveguides may be dynamically controlled with all-optical means. Through computer simulations, we showed that a waveguide modulator whereby an optical vortex soliton induced a cylindrical waveguide within a self-defocusing medium and another beam was used to modulate the refractive index within a small region of the waveguide. On the fundamental understanding of dark solitons with two transverse degrees of freedom, we have studied polarized OVS's. We have found that only circularly polarized OVS's are stable and other polarized OVS's are susceptible to long period of spatial perturbation. What is remarkable here is that an unstable polarized OVS exhibits striking decay dynamics. To perform the above research, we have developed a program for solving vector nonlinear Schrödinger Equation. The results of this project have been shared with Dr. G. A. Swartzlander, Jr., O. N. T. Postdoctoral Fellow, and Dr. A. J. Campillo, Senior Research Scientist, both of the Naval Research Laboratory and have been presented in various national conferences.

2a. Polarized optical vortex solitons

Linear vortices are already known to play a fundamental role in the scattering of radiation and in waveguides, where the vortex is characterized by the separable function of the azimuthal coordinate: $\exp(iM\phi)$, and $M = \pm 1, \pm 2, \dots$ is the so-called topological charge. Cylindrical symmetry allows various polarization modes, e. g., transverse-electric, transverse-magnetic, and circular. One may expect nonlinear optical vortices, which also have cylindrical symmetry, to exhibit interesting polarization dynamics owing to the symmetry-breaking effect of an anisotropic light-induced refractive index change. We have discovered that out of six characteristically different polarized vortex states, only the circular polarized waves are stable. What is also remarkable is that the decay dynamics of the other five modes exhibit two striking phenomena: (1) the initial vortex core vanishes and then re-emerges with the opposite topological charge, and (2) additional pairs of vortices are generated (while preserving the net topological charge).

The propagation of polarized light in a nonlinear refractive medium is described by the paraxial wave equation,

$$i2k_0 \partial \vec{E} / \partial z + \left[\partial^2 / \partial x^2 + \partial^2 / \partial y^2 \right] \vec{E} + k_0^2 \vec{D}^{NL} / n_0 = 0 \quad (1a)$$

where n_0 is the linear refractive index (assumed isotropic), k_0 is the wave number inside the medium directed along the z -axis, and \vec{E} is the electric field. The electric displacement for a Kerr nonlinear medium is given by,

$$\vec{D}^{NL} = 2n_2 \epsilon_0 [A(\vec{E} \cdot \vec{E}^*) \vec{E} + B(\vec{E} \cdot \vec{E}) \vec{E}^*] \quad (1b)$$

where n_2 is the nonlinear refractive index coefficient for linearly polarized light field, and, for a lossless medium, $B = 1 - A$ is a measure of the light-induced anisotropy. In a self-defocusing medium, n_2 is negative, and the intensity-dependent refractive index is given by $n = n_0 - \Delta n$, where $\Delta n \equiv |n_2| \vec{E}(x, y, z) \cdot \vec{E}^*(x, y, z)$.

It is naturally convenient to decomposed \vec{E} into two circularly polarized components: $\vec{E} = \vec{V}(x, y) \exp(-i2Z)$, $\vec{V} = [E_R u_R(x, y) \hat{e}_R + E_L u_L(x, y) \hat{e}_L]$, where $\hat{e}_R = 2^{-1/2}(\hat{x} - i\hat{y})$ and $\hat{e}_L = 2^{-1/2}(\hat{x} + i\hat{y})$ are the unit vector for the right (R) and left (L) circularly polarized components respectively, E_R and E_L are the values of the background field amplitudes of each circularly polarized component, \hat{x} and \hat{y} are transverse unit vectors, u_R and u_L are the normalized amplitude profile of the field, $Z = z/L_{\parallel}$ is the normalized propagation distance [$L_{\parallel} = 2k_0^{-1}(n_0/\Delta n_{\infty})$], and Δn_{∞} is the index change attributed to the background field. We ignore small field components in the \hat{z} direction that occurs for transverse magnetic waves. From Eq. (1a) and (1b), two coupled nonlinear Schrödinger (NLS) equations are obtained:

$$2u_R + i\partial u_R / \partial Z + \nabla_{\perp}^2 u_R - 2 \left[(1-B)|u_R|^2 \eta_R^2 + (1+B)|u_L|^2 \eta_L^2 \right] u_R = 0 \quad (2a)$$

$$2u_L + i\partial u_L / \partial Z + \nabla_{\perp}^2 u_L - 2 \left[(1-B)|u_L|^2 \eta_L^2 + (1+B)|u_R|^2 \eta_R^2 \right] u_L = 0 \quad (2b)$$

where $\nabla_{\perp}^2 = \partial^2 / \partial X^2 + \partial^2 / \partial Y^2$, $\eta_R = E_R / |\vec{E}_{\infty}|$ and $\eta_L = E_L / |\vec{E}_{\infty}|$ give the fraction of energy in each polarization state ($|\vec{E}_{\infty}| = (E_R^2 + E_L^2)^{1/2}$), and $(X, Y) = (x/L_{\perp}, y/L_{\perp})$ are the normalized transverse coordinates ($L_{\perp} = 2k_0^{-1}(n_0/\Delta n_{\infty})^{1/2}$).

The scalar NLS equation may admit vortex solutions of the form $u_M(\rho, \phi) = \psi(\rho) \exp(iM\phi)$, where $\rho = (X^2 + Y^2)^{1/2}$ and ϕ is the azimuthal coordinate in the transverse plane. Now we can

inquire whether polarized waves allow stable vortex modes in a self-defocusing medium. An arbitrary vortex field, V , may be composed by superimposing a pair of right- and left-handed polarized vortices of topological charges M and N , e. g., $\vec{V} = E_R u_M \hat{e}_R + E_L u_N \hat{e}_L$. From this infinite selection of modes, we examine the most fundamental, namely, those with unit charge ($M, N = \pm 1$) whose polarization components are either equal in amplitude ($E_R = E_L$) or dominated by one (e.g., $E_L = 0$). This allows us to identify six polarized vortex modes of special significance: “linear” $\vec{V}_{Lin} = E_R u_M (\hat{e}_R + \hat{e}_L)$, “circular” $\vec{V}_{Cir} = E_R u_M \hat{e}_R$ or $E_L u_M \hat{e}_L$, “radial” $\vec{V}_{Rad} = E_R (u_1 \hat{e}_R + u_{-1} \hat{e}_L) = 2^{1/2} E_R \psi(\rho) \hat{r}$, “radial-compliment” $\vec{V}_{Rad} = E_R (u_{-1} \hat{e}_R + u_1 \hat{e}_L)$, “ ϕ ” $\vec{V}_\phi = E_R (u_1 \hat{e}_R - u_{-1} \hat{e}_L) = 2^{1/2} \exp(-i\pi/2) E_R \psi(\rho) \hat{\phi}$, and “ ϕ -compliment” $\vec{V}_\phi = E_R (u_{-1} \hat{e}_R - u_1 \hat{e}_L)$. Note that for circularly polarized waves $\Delta n_\infty = A n_2 E_{R,L}^2$ whereas other polarizations under investigation have $\Delta n_\infty = n_2 (E_R^2 + E_L^2)$. While many other modes involving higher or mixed charges and elliptical polarization will not be treated here, one may expect that the results of the following stability analyses can provide enough insight to allow at least qualitative descriptions for other cases.

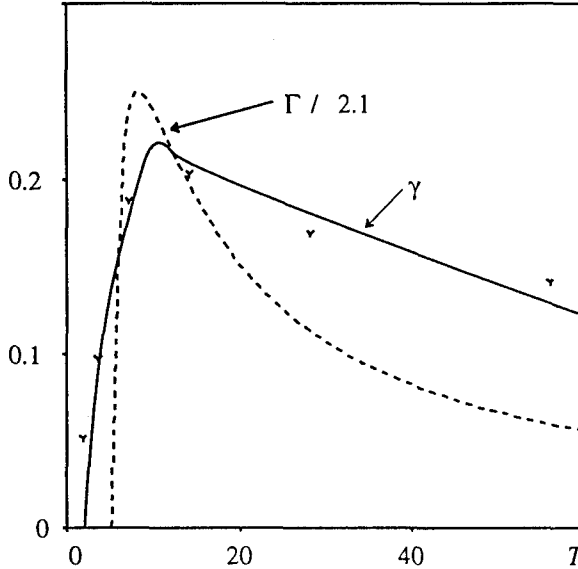


Fig. 1. Instability growth rates as a function of normalized modulation period, T . The calculated initial growth rate, Γ , is shown for radial perturbations (dashed). The average growth rate, γ , is calculated numerically (data points) and fitted (solid) to Eq. (4) with a single parameter. For both cases we set the radial vortex core size to $r_0 = 25 \mu\text{m}$, the modulation amplitude to $\varepsilon = 0.1$, and material parameters to $\Delta n_\infty / n_0 = 2 \times 10^{-5}$ and $B = 1/3$. The initial growth rate, Γ , is roughly 2.1 times the average rate, γ .

To determine whether these six vortex modes are stable, we perform a linear stability analysis for each of them. We find out that the propagation constant Γ for these perturbed modes can be expressed as a function of normalized spatial frequency $k_\perp \equiv 2\pi/T$ or normalized spatial period T (in physical unit, the period is $t = T L_\perp$):

$$\Gamma = \pm i k_\perp (k_\perp + 2A)^{1/2}, \pm i (2 + k_\perp^2)^{1/2} \quad \text{for circular polarization} \quad (3a)$$

$$\text{and } \Gamma = \pm i k_\perp (4 + k_\perp^2)^{1/2}, \pm k_\perp (4B - k_\perp^2)^{1/2} \quad \text{for the remaining five polarizations.} \quad (3b)$$

Note that only the last expression in Eq. (3b) may be real under certain conditions. Therefore the circular polarization cases are stable to radial perturbations, owing to the fact that Γ is always imaginary. This is not a surprise since the NLS equation for the circular polarization case is

identical in form to the scalar NLS equation, the latter has already been found to be stable (both in theory and practice).

On the other hand, the remaining five polarization states admit an exponential growth of the perturbation. This may have been expected since it is known that elliptically polarized plane waves are also unstable under cross-phase modulation. From the last expression of Eq. (3b), we notice that the vortex is unstable whenever $k_{\perp} < k_{cr} \equiv 2B^{1/2}$ (i. e., whenever the physical period of modulation exceeds $t_{cr} = \pi L_{\perp} B^{-1/2}$). The growth rate of instability in the last expression of Eq. (3b) can be written in terms of T and T_{cr} :

$$\Gamma = 4B(1 - T_{cr}/T)^{1/2} T_{cr}/T. \quad (4)$$

The growth of the instability peaks at $T_{opt} = \sqrt{2} T_{cr} = 7.7$, as depicted in Fig. 1, is a plot of Eq. (4) for $B = 1/3$. Since perturbations used in the linear stability analysis preserve the cylindrical symmetry of the initial vortices and maintain zero field at the center of the vortices, they may not be the dominant mode causing the strongest instability.

We have used numerical methods to perform a more rigorous nonlinear instability analysis which allows not only the determination of growth rates, but also qualitative information on the decay process. For numerical convenience, we have examined a sinusoidal perturbation along the x-axis:

$$\vec{E} = E_R \left[u_M + \varepsilon \sin(k_{\perp} X) \right] \hat{e}_R + E_L \left[u_{\pm M} - \varepsilon \sin(k_{\perp} X) \right] \hat{e}_L \quad (5)$$

where we set the amplitude of modulation to $\varepsilon = 0.1$, and $M = \pm 1$. The perturbation modulates each of the circular components with the same frequency k_{\perp} , but π out of phase.

The decay dynamics from these modulations have been investigated by numerically solving Eq. (2) with the beam propagation method. A vortex (with core radius $r_0 \equiv L_{\perp}/\sqrt{2} = 25 \mu\text{m}$) having one of the five polarizations under consideration is subjected to periodic modulations ($t = 1.0 \text{ mm}$) on a 512×512 transverse grid, and a moderate nonlinearity of $\Delta n_{\infty}/n_0 = 2.0 \times 10^{-5}$ is assumed in a medium with $B = 1/3$ at a wavelength of $1 \mu\text{m}$. Under these conditions the nonlinear scaling size, L_{\perp} , corresponds to 5 pixels in our computations. The dynamic features for each of the five polarizations are very similar. Therefore, we present here only calculations for azimuthal polarization, \vec{V}_{ϕ} . The intensity and phase evolution are shown in Figs. 2 and 3, respectively, at different propagation distances. The gray-scale palette maps small values of intensity or phase (modulus 2π) into dark tones; a logarithmic (linear) scale is used for intensity (phase). These cross-sectional images span only 12% of the computation grid area. The left-handed circular component resembles the corresponding right circular component after it is rotated in the x-y plane by 90° about the center of the initial vortex; thus it is not shown.

As expected, these figures show that the perturbation has clearly destroyed the circular symmetry of the vortex located at the origin ($x=y=0$). Remarkable and unexpected dynamics occurs as the beam propagates. For example, in (B) of Figs. 2 and 3, the vortex core has completely vanished! Instead, a π phase front resembling a dark soliton segment is formed in the vicinity of the origin. (This may be interpreted as a topological vortex that osculates the x-y plane, and thus, does not actually violate the conservation of charge principle.) Also note that although the vortex itself has disappeared, the global phase continues to have the same sense of rotation (counter-clockwise). As the beam proceeds further to (C) in Figs. 2 and 3, three vortices develop, with the central one at the origin having the opposite topological charge of the initial vortex, i.e., its sense of rotation has reversed! The two "new" vortices have the same charge, and thus, the net vorticity ($\sum M_i$) is the same as the initial condition. (This may be interpreted as a single vortex filament or "string" passing through the transverse plane three times.) The

complexity appears to develop more quickly as more vortices are generated, as evidenced in frame (D). In this case, the vortex filament osculates the x-y plane in several locations; a short distance further, seven vortices appear. As the beam proceeds further, one may expect that nonlinear optical turbulence will eventually develop.

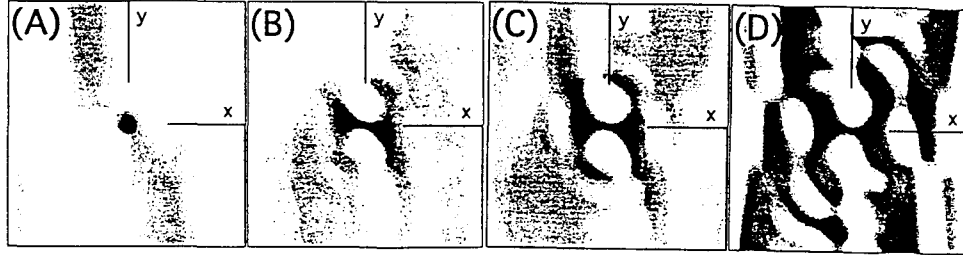


Fig. 2. Transverse intensity profiles of the right-hand circularly polarized component of a transverse electric vortex mode (core radius $R_0 = 25\mu m$, nonlinearity $\Delta n_{NL}/n_0 = 2.0 \times 10^{-5}$) perturbed by a modulation with an amplitude 10% of the beam intensity and a period of 1mm at $Z=0$ (0 cm), 5.9 (9.4 cm), 7.5 (11.9 cm), and 11 (18.2 cm) in (A), (B), (C) and (D) respectively.

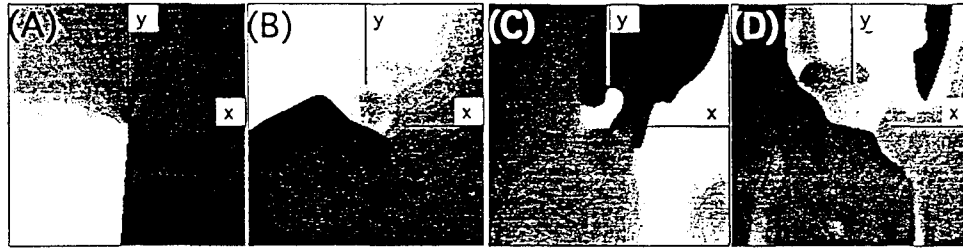


Fig. 3. Same as Fig 3, except phase is shown (black corresponds to zero phase, and white to 2π). The global vorticity has a counter-clockwise sense. The vortex core is absent in (B) and (D); instead π phase fronts exist and the corresponding intensity profiles show dark soliton segments.

In general, the vector vortex dynamics may be interpreted as two co-linear vortex filaments of opposite circular polarization that unravel (owing to cross-phase modulation) as they propagate through the self-defocusing medium. If one of the polarized filaments is absent or if the nonlinear coupling is "turned off", stable propagation is expected from the scalar theory. If both polarization modes are present, their coupling effect will depend on the relative power in each mode, as well as the material parameter, B . For a given value of B , we expect the strongest coupling when the two modes have equal strength, i. e., the five unstable modes described above. Based on these results, one may expect spontaneous vortex formations that appear as closed loops in three dimensions, which may occur when a plane wave of arbitrary polarization (though not circular) is perturbed at different periods along both the x- and y-axes.

Possible impact of these research results is a potential scheme for making a polarization sensitive all-optical switch. For example, a light-induced waveguide is formed in a right circularly polarized beam with an OVS. Another low intensity probe beam with different color can be guided along this light-induced waveguide along the dark core of the OVS. If another gate probe with the same frequency but different polarization is introduced to the guiding beam, the polarized OVS may become unstable. As a result, the guided probe beam will be switched off owing to the disintegration of the light-induced wavelength.

2b. All-optical modulator

Besides the all-optical switch making use of polarization effect, another idea is to exploit the property that the size of the light-induced fiber ("light pipe") changes with the intensity of the bright region of the guiding beam. Since an OVS has a core radius $r_0 \equiv L_{\perp}/\sqrt{2}$ where $L_{\perp} = 2k_0^{-1}(n_0/\Delta n_{\infty})^{1/2}$, $n_{\infty} = n_2|E_{\infty}|^2$ and E_{∞} is the background field, the radius of an OVS is proportional to $1/|E_{\infty}|$. By squeezing and relaxing the "light pipe", we can encode information into the flow of the wave. Therefore, we can achieve this by modulating the intensity of the bright region. We can easily introduce modulations to the guiding beam by sending a control beam to it. Consequently, whatever signal is present in the control beam will be transported to the guided light. Fig. 4 displays the relations among the guiding beam (with an OVS), the guided probe beam and the gate beam as well as the increase in Δn_{∞} owing to the gate beam.

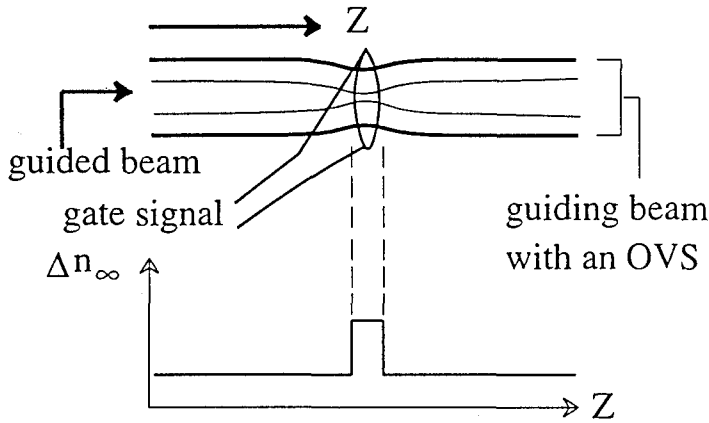


Fig. 4. All-optical modulator based on the intensity-dependent core size of an OVS. The core size as well as the probe beam guided by the OVS inside a guiding beam is modulated by the gate signal

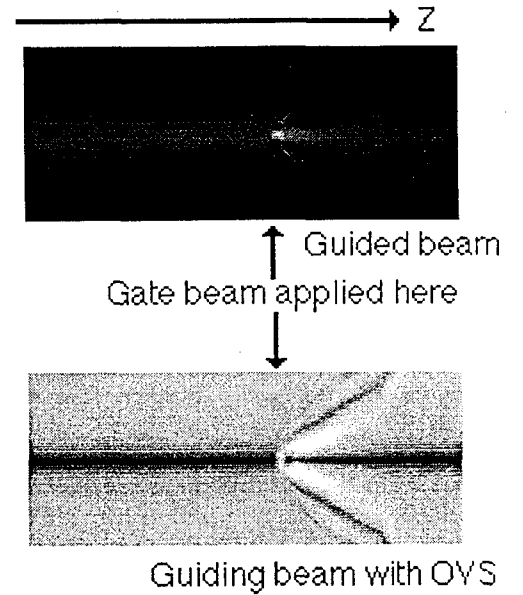


Fig. 5. Pinch-off of the light-induced waveguide (OVS). Observe that after passing through the gate beam, the guided beam (the upper picture) radiates and dissipates while the OVS in the guided beam (the lower picture) returns to original size.

Owing to the reduction in the the dimension of the waveguide, portion of the guided beam must radiate out. If the change in Δn_{∞} is abrupt, a substantial percentage of the guided beam can escape from the waveguide. Our prediction is confirmed by results of numerical simulations of a guiding beam with $r_0 = 25\mu m$ and nominal $\Delta n_{\infty} = 2 \times 10^{-5}$. Fig. 5 shows the topview (along the propagation direction) of a guided beam with intensity 0.001 of the OVS's background intensity and a guiding beam with an OVS ("light pipe") in one of our computer simulations where a gate beam is directed to the guiding beam perpendicularly. The gate beam causes Δn_{∞} to increase to 6×10^{-4} . The cross section of "light pipe" shrinks abruptly for the portion of guiding beam under the influence of the gate beam, and relaxes to original size when the effect of the gate beam diminishes. The shrinkage of the "light pipe" blocks off a portion of the

power of the guided beam. As a result, the guided beam is invisible after traversing the gate beam. This is analogous to the pinch-off effect happening in the channel ("light pipe") of a field effect transistor.

We have also examined the dynamics of the guided pulse under different levels of gate beam power. Fig. 6 shows the simulation results of applying various gate signal intensities around the longitudinal distance between $9.1Z_0$ and $9.2Z_0$. The different gate intensities result in Δn_∞ varying from 10^{-4} to 8×10^{-4} . As the gate power increases, the guided power within the longitudinal region enclosed by the gate beam jumps up. After the longitudinal region, background intensity of the guiding beam returns to nominal value and so does the OVS's core radius. This change in the waveguide geometry leads to the leakage of the guided radiation and the drop in guided radiation within a circular aperture of radius $=2r_0$. Fig. 6 demonstrates that the transmission coefficient of the guided beam, which measures the amount of radiation passing through the circular aperture, reaches a steady and lower value at a distance of $9.4Z_0$, i.e. $0.2Z_0$ away from the gate beam region. This steady value of transmission coefficient after the gate beam decreases almost linearly as the change in nonlinear refractive index increases.

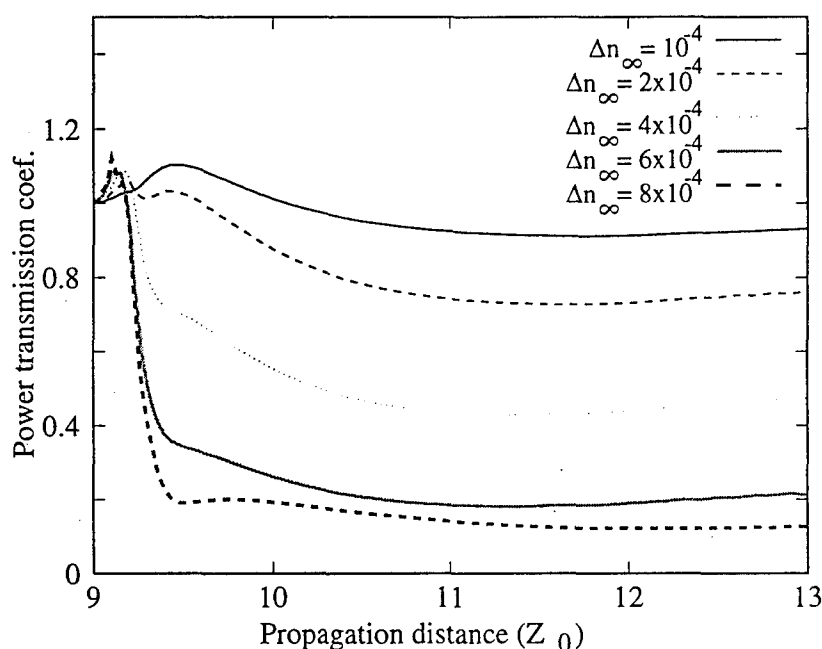


Fig. 6. Transmission coefficient of the guided beam through a aperture with radius $=2r_0$ versus the propagation distance. The gate beam locates at the distance between $9.1Z_0$ and $9.2Z_0$.

2c. Propagation program for vector nonlinear Schrödinger equation

We have extended a program, which was developed to simulate 3 dimensional nonlinear wave propagation, to solve the vector nonlinear Schrödinger equation in Eq. (2). We have ported the program to Cray C-90 supercomputer. As a result, it takes less than 3 minutes of cpu time to simulate optical wave propagation to a distance of three diffraction length. The results of each run are usually stored as many frames of pictures with typical resolution of 512×512 pixels in HDF (Hierarchical Data Format) which is developed by the National Center for Supercomputing Applications. This image storage format reduces the storage space

substantially. Various programs have been implemented for processing the resulting images and format conversion.

The core of the nonlinear propagation code is based on the split-step method (or beam propagation method). This numerical technique was originally used to solve the scalar nonlinear Schrödinger equation at a given transverse plane by splitting each numerical step into a linear half step without nonlinearity and a nonlinear half step without diffraction. We modify it here to solve Eq. (2) which consists of two coupled nonlinear Schrödinger equations. The nonlinear phase correction is computed in real space whereas the linear phase contribution from diffraction is computed in the Fourier transform plane. Therefore, the electric field is transformed into Fourier components during the linear step and is transformed back into real space during the nonlinear step. These transformations are carried out by the fast 2D FFT (Fast Fourier Transformation) on the Cray C-90. Since the version of FFT on the Cray supercomputer are highly optimized, results can be obtained from simulations using the code in a very short time.

The listing of the new nonlinear propagation program can be found in Section 3. This code can be run on other UNIX workstations with the IMSL mathematical library and the HDF library installed.

3. Listing of the nonlinear propagation program

We list here the beam propagation program (called non2beam.F) for solving vector nonlinear Schrödinger equation in Fig. 7. Non2beam.F can simulate nonlinear propagation of the circularly polarized components of a beam with arbitrary polarization or nonlinear co-propagation of 2 beams with different frequencies. This program requires an input data file which is listed in Fig. 8 (called non2beam.dat) which contains data for the nonlinear propagation. On a Cray C-90 supercomputer with UNICOS system, the program is compiled by issuing the following command:

```
cf77 -o run non2beam.F -ldf
```

where the “-ldf” flag is for the library of HDF subroutines (at the Pittsburgh Supercomputing Center, this library is put in /usr/local/lib/libdf.a which is in the automatic search path of cf77. Cray supercomputers at other locations may put it under other directories.) On a typical UNIX workstation, the program can be compiled by issuing the following command:

```
f77 -O -o non2beam non2beam.F -ldf -limsl10
```

where the flags “-ldf” and “-limsl10” are for the libraries of HDF subroutines and IMSL subroutines, e.g. f77 looking for libraries with names libdf.a and libimsl10.a under its regular search path. After compilation, the program can be executed with the following command:

```
non2beam < non2beam.dat >> non2beam.dat
```

This command appends output information to the input file input.dat. Other output files are specified in non2beam.dat.

Fig. 7. Listing of the nonlinear propagation program non2beam.F

```
program non2beam
c
c Program for solving 3D nonlinear Schrodinger equation
c with beam propagation method for two beam
c
c define polar for decomposed a beam into circular components
c
      parameter (mx=512,md=mx+1,my=512)
#ifdef cray
      parameter (ntable=100+2*(mx+my), nwork=4*mx*my)
#else
#ifdef sgi
      parameter (ntbl=30+my+mx,ncp=2*mx)
#else
      parameter (ntbl=15+4*mx,ncp=2*mx)
#endif
#endif
      complex u(md,my),phX(mx),phY(my),cpxi,prox,proy,dem,optnon
      complex phlin(md,my), temp, tmpo(8), u2(md,my), temp2
#ifdef cray
```

```

        real table(ntable), work(nwork)
#else
#ifdef sgi
        complex wff1(ntbl)
        real cpy(ncp)
#else
        real wff1(ntbl), wff2(ntbl), cpy(ncp)
#endif
#endif
        complex work(mx,my)
#endif
        integer iflag
        character filnam*18,filnam3*18,ch*20,ans1*1,ans2*1,filnam4*18
        character filnam2*18, filnam23*18, filnam24*18
c
c Default value for some of the inputs
c
        data wlam,wid,eps,sclx,scly,z1 /1.e-6, 1.e-4, 1.e-3,1., 1.,0./
        data isig,ipr,nfrm /-1,5,1/ filnam3 /'ft.d'/ filnam4 /'ifr.hdf'/
        data zinit /0./ refn0/1./ filnam23 /'ft2.d'/ filnam24 /'ifr2.hdf'/
c
c Skip the first line in the data file
c
        2 format(i10)
        3 format(e20.13)
        read(5,*)
        read(5,1) filnam
        read(5,1) filnam2
        read(5,1) ans1
        if (ans1 .eq. 'y') then
            read(5,1) ch
            if (ch .ne. " ") read(ch,2) nfrm
            read(5,1) ch
            if (ch .ne. " ") read(ch,3) zinit
        else
            read(5,*)
            read(5,*)
        endif
        read(5,5) e2n
        5 format(70x,e20.13)
        iflag = 0
        if (abs(e2n) .lt. 1.e-30) iflag=1
        read(5,1) ch
        if (ch .ne. " ") read(ch,3) wlam
        read(5,1) ch
        if (ch .ne. " ") read(ch,3) wid
c
c Get the scale factor for the x and y range
c
        read(5,1) ch
        if (ch .ne. " ") read(ch,3) sclx
        read(5,1) ch
        if (ch .ne. " ") read(ch,3) scly
        read(5,1) ch
        1 format(70x,a)
        if (ch .ne. " ") filnam3=ch
#ifdef cray
        open(unit=3,file=filnam3,status='unknown',access='direct',

```

```

1 recl=128,form='unformatted')
#else
    open(unit=3,file=filnam3,status='unknown',access='direct',
1 recl=16,form='unformatted')
#endif
    read(5,1) ch
    if (ch .ne. " ") filnam23=ch
#ifdef cray
    open(unit=7,file=filnam23,status='unknown',access='direct',
1 recl=128,form='unformatted')
#else
    open(unit=7,file=filnam23,status='unknown',access='direct',
1 recl=16,form='unformatted')
#endif
    pi = 4.*atan(1.)
    pi2 = 2.*pi

c
c e2n --  $n^2|E_0|^2/n_0$ 
c refn0 -- linear refractive index
c isig -- sign of nonlinearity
c xrag -- positive range of x in meter, [-xrag, xrag]
c yrag -- positive range of y in meter, [-yrag, yrag]
c zrag -- final propagation distance z/zo
c wlam -- wavelength in meter
c wid -- beam width in meter
c eps -- error control
c pratio -- power ratio of beam 2 to beam1
c couple -- coupling coef. between beam1 and beam2 (1+b)/a
c
    read(5,1) ch
    if (ch .ne. " ") read(ch,3) refn0
    read(5,1) ch
    if (ch .ne. " ") read(ch,2) isig
    read(5,1) ch
    if (ch .ne. " ") read(ch,3) zrag
#ifdef cray
    open(unit=2,file=filnam,status='old',form='unformatted',
1 access='direct',recl=128)
#else
    open(unit=2,file=filnam,status='old',form='unformatted',
1 access='direct',recl=16)
#endif
    read(2,rec=1) nx, ny
    xrag = sclx*wid*0.5*sqrt(nx*pi)
    yrag = sclx*wid*0.5*sqrt(ny*pi)
    wnum = 2.*pi/ wlam
    xlen = xrag/wid
    ylen = yrag/wid
    zo = refn0*pi*wid**2/wlam
    zdist = zrag * 0.25
    write(6,6) xrag,xrag, yrag, yrag,zdist,zo,wnum
6 format(" x/wid range [-",e14.7," m, ",e14.7," m]; ",
1 "y/wid range [-",e14.7," m, ",e14.7," m]"/" z/4/Zo is "
1 ,f10.3," (Zo ",e14.7," m); wavelength number - ",e14.7
1 , " 1/m")
    read(5,1) ch
    if (ch .ne. " ") read(ch,3) eps

```

```

    ntot = nx*ny
    nx2 = nx + 2
    ny2 = ny + 2
    fnx = nx
    fny = ny
    nxd2 = nx / 2
    nyd2 = ny / 2
    cpxi = (0.,1.)
    delx = 2.*xlen/nx
    dely = 2.*ylen/ny
c
c  delz is small enough so that the linear phase <= pi/2
c
    delz = min(1./(float(nx)/sclx**2+float(ny)/scly**2),eps)
    iter = int(zdist/delz) + 1
    write(6,20) iter
20 format(' number of iterations estimated = ', i5)
    read(5,1) ans2
    if (ans2 .eq. 'y') then
        read(5,1) ch
        if (ch .ne. " ") filnam4 = ch
        read(5,1) ch
        if (ch .ne. " ") filnam24 = ch
c
c  define reference level
c
    rlv = 0.5
    else
        read(5,*)
    endif
    read(5,1) ch
    if (ch .ne. " ") read(ch,2) ipr
    majlp = iter/ipr
    if (mod(iter,ipr) .ne. 0) then
        majlp = majlp + 1
        iter = majlp*ipr
    endif
c
c  Get beginning distance and ending distance
c
    read(5,1) ch
    if (ch .ne. " ") read(ch,3) z1
    read(5,5) pratio
    read(5,5) couple
    write(6,27) z1, e2n, pratio, couple
27 format(" z1 = ",e14.7, " nonlinearity = ",e14.7/
1 " power2/power1 = ",e14.7," coupling coef. = ",e14.7)
    zst = z1 * 0.25
    delz = zdist/iter
    write(6,25) nx,ny,iter,delz,ipr,eps
25 format(1x,i5,' x ',i5,' x-y dimension, ',
1 ' number of iteration ',i5//' dz ',e14.7' write ',
1 i5,' frames with tolerance < ',e14.7)
c
c  compute linear propagation in x direction
c
    if (iflag .eq. 1) then

```

```

    prox = cpxi*delz*float(majlp)*(pi/xlen)**2
    else
    prox = cpxi*delz*(pi/xlen)**2*0.5
    endif
    phX(1) = 1.
    phX(nxd2+1) = exp(prox*(nxd2)**2)
    do 30 i = 2, nxd2
        phX(i) = exp(prox*(i-1)**2)
30 phX(nx2-i) = phX(i)
c
c compute linear propagation in y direction
c
    if (iflag .eq. 1) then
    proy = cpxi*float(majlp)*delz*(pi/ylen)**2
    else
    proy = cpxi*delz*(pi/ylen)**2*0.5
    endif
    phY(1) = 1.
    phY(nyd2+1) = exp(proy*(nyd2)**2)
    do 40 i = 2, nyd2
        idx = i - 1
        phY(i) = exp(proy*idx**2)
40 phY(ny2-i) = phY(i)
    do 50 j = 1, ny
        do 50 i = 1, nx
50 phlin(i,j) = phX(i)*phY(j)
        zprop = 0.
        dntot = 1. / float(ntot)
c
c recompute phX phY for later use
c
    prox = cpxi*(pi/xlen)**2
    phX(1) = 0.
    phX(nxd2+1) = prox*(nxd2)**2
    do 53 i = 2, nxd2
        phX(i) = prox*(i-1)**2
53 phX(nx2-i) = phX(i)
    proy = cpxi*(pi/ylen)**2
    phY(1) = 0.
    phY(nyd2+1) = proy*(nyd2)**2
    do 51 i = 2, nyd2
        idx = i - 1
        phY(i) = proy*idx**2
51 phY(ny2-i) = phY(i)
c
c get initial field distribution
c
    idx2 = ntot*(nfrm-1)/8 + 1
    do 10 i = 1, ntot-7, 8
        idx2 = idx2 + 1
10 read(2,rec=idx2) (u(mod(i+k,nx)+1,(i+k)/nx+1),k=-1,6)
    close (2)
#ifdef cray
    open(unit=4,file=filnam2,status='old',form='unformatted',
1 access='direct',recl=128)
#else
    open(unit=4,file=filnam2,status='old',form='unformatted',

```



```

1  access='direct',recl=16)
#endif
    idx22 = ntot*(nfrm-1)/8 + 1
    do 310 i = 1, ntot-7, 8
        idx22 = idx22 + 1
        310 read(4,rec=idx22) (u2(mod(i+k,nx)+1,(i+k)/nx+1),k=-1,6)
        close (4)
#ifdef cray
#ifdef sgi
        call cfft2di(nx,ny,wff1)
#endif
#endif
        if (ans1 .eq. 'n') then
c
c  split2 routine
c
        do 210 i = 1, ny
            do 215 j = 1, nxd2
                temp = u(j,i)
                temp2 = u2(j,i)
                jnx2 = j+nxd2
                u(j,i) = u(jnx2,i)
                u2(j,i) = u2(jnx2,i)
                u2(jnx2,i) = temp2
            215 u(jnx2,i) = temp
        210 continue
            do 220 j = 1, nyd2
                do 225 i = 1, nx
                    temp = u(i,j)
                    temp2 = u2(i,j)
                    jny2 = j+nyd2
                    u(i,j) = u(i,jny2)
                    u2(i,j) = u2(i,jny2)
                    u2(i,jny2) = temp2
                225 u(i,jny2) = temp
            220 continue
c
c  end split2 routine
c
#ifdef cray
        call cfft2d(1,nx,ny,1.,u,1,md,u,1,md,table,ntable,
1        work,nwork)
        call cfft2d(1,nx,ny,1.,u2,1,md,u2,1,md,table,ntable,
1        work,nwork)
#else
#ifdef sgi
        call cfft2d(-1,nx,ny,u,md,wff1)
        call cfft2d(-1,nx,ny,u2,md,wff1)
#else
        call f2t2d(nx,ny,u,md,u,md,wff1,wff2,work,cpy)
        call f2t2d(nx,ny,u2,md,u2,md,wff1,wff2,work,cpy)
#endif
#endif
        endif
c
c  Propage z1 distance in front of the medium
c

```

```

        if (abs(zst) .gt. 1.e-30) then
        do 70 i = 1, ny
            do 70 j = 1, nx
                u2(j,i) = u2(j,i) * exp((phX(j)+phY(i))*zst)
70      u(j,i) = u(j,i) * exp((phX(j)+phY(i))*zst)
            endif
        #ifdef polar
            optconst = -2.*(wnum*wid)**2*isig*e2n*refn0/(1.+couple)
        #esle
            optconst = -4.*(wnum*wid)**2*isig*e2n*refn0/(1.+couple)
        #endif
        phzinit = optconst*0.25*zinit
c
c  write the initial fields
c
        write(3,rec=1) nx, ny
        write(7,rec=1) nx, ny
        koo3 = 1
        if (ans2 .eq. 'n') then
            do 65 i = 1, ntot-7, 8
                koo3 = koo3 + 1
                write(7,rec=koo3) (u2(mod(i+k,nx)+1,(i+k)/nx+1),k=-1,6)
65      write(3,rec=koo3) (u(mod(i+k,nx)+1,(i+k)/nx+1),k=-1,6)
            else
        #ifdef cray
            call cfft2d(-1,nx,ny,dntot,u,1,md,u,1,md,table,ntable,
1          work,nwork)
            call cfft2d(-1,nx,ny,dntot,u2,1,md,u2,1,md,table,ntable,
1          work,nwork)
        #else
        #ifdef sgi
            call cfft2d(1,nx,ny,u,md,wff1)
            call cscal2d(nx,ny,dntot,u,md)
            call cfft2d(1,nx,ny,u2,md,wff1)
            call cscal2d(nx,ny,dntot,u2,md)
        #else
            call f2t2b(nx,ny,u,md,u,md,wff1,wff2,work,cpy)
            call f2t2b(nx,ny,u2,md,u2,md,wff1,wff2,work,cpy)
            do 279 i = 1, ny
                do 279 j = 1, nx
                    u2(j,i) = dntot*u2(j,i)
279      u(j,i) = dntot*u(j,i)
        #endif
        #endif
            call hdf_img2(filnam4,phzinit,rlv,u,md,nx,ny,0,work,ntot,2)
            call hdf_img2(filnam24,phzinit,rlv,u2,md,nx,ny,0,work,ntot,2)
        #ifdef cray
            call cfft2d(1,nx,ny,1.,u,1,md,u,1,md,table,ntable,
1          work,nwork)
            call cfft2d(1,nx,ny,1.,u2,1,md,u2,1,md,table,ntable,
1          work,nwork)
        #else
        #ifdef sgi
            call cfft2d(-1,nx,ny,u,md,wff1)
            call cfft2d(-1,nx,ny,u2,md,wff1)
        #else
            call f2t2d(nx,ny,u,md,u,md,wff1,wff2,work,cpy)

```

```

        call f2t2d(nx,ny,u2,md,u2,md,wff1,wff2,work,cpy)
#endif
#endif
        endif
        energ = 0.
        energ2 = 0.
        do 60 i = 1, ny
            do 60 j = 1, nx
                energ2 = energ2 + real(u2(j,i)*conjg(u2(j,i)))
        60 energ = energ + real(u(j,i)*conjg(u(j,i)))
        write(6,*) zprop, energ*dntot, energ2*dntot
c
c  Start the split step method: linear part*nonlinear part*linear part
c
        roptnon = optconst*delz
        optnon = cpxi*roptnon
        roptnon = float(majlp) * roptnon
        do 120 i = 1, ipr
            if (iflag .eq. 1) then
                do 113 k = 1, ny
                    do 113 jj = 1, nx
                        u2(jj,k) = u2(jj,k) * phlin(jj,k)
113          u(jj,k) = u(jj,k) * phlin(jj,k)
                zprop = zprop + delz*majlp
            else
                do 110 j = 1, majlp
                    zprop = zprop + delz
c
c  linear half step
c
                do 103 k = 1, ny
                    do 103 jj = 1, nx
                        u2(jj,k) = u2(jj,k) * phlin(jj,k)
103          u(jj,k) = u(jj,k) * phlin(jj,k)
            #ifdef cray
                call cfft2d(-1,nx,ny,dntot,u,1,md,u,1,md,table,ntable,
1          work,nwork)
                call cfft2d(-1,nx,ny,dntot,u2,1,md,u2,1,md,table,ntable,
1          work,nwork)
            #else
            #ifdef sgi
                call cfft2d(1,nx,ny,u,md,wff1)
                call cscal2d(nx,ny,dntot,u,md)
                call cfft2d(1,nx,ny,u2,md,wff1)
                call cscal2d(nx,ny,dntot,u2,md)
            #else
                call f2t2b(nx,ny,u,md,u,md,wff1,wff2,work,cpy)
                call f2t2b(nx,ny,u2,md,u2,md,wff1,wff2,work,cpy)
                do 179 ii = 1, ny
                    do 179 ji = 1, nx
                        u2(ji,ii) = dntot*u2(ji,ii)
179          u(ji,ii) = dntot*u(ji,ii)
            #endif
        #endif
c
c  nonlinear full step
c

```

```

        do 100 k = 1, ny
            do 100 jj = 1, nx
                temp = u(jj,k)*conjg(u(jj,k))
                temp2 = u2(jj,k)*conjg(u2(jj,k))*pratio
                u2(jj,k) = u2(jj,k)*cexp(optnon*(temp2+couple*temp))
100         u(jj,k) = u(jj,k)*cexp(optnon*(temp+couple*temp2))
c         dem = (1.,0.d0) + optnon*u(k)*conjg(u(k))
c 100     u(k) = u(k) * ((-1.,0.) + (2.,0.)/dem)**2
c
c linear half step
c
#ifdef cray
    call cfft2d(1,nx,ny,1.,u,1,md,u,1,md,table,ntable,
1        work,nwork)
    call cfft2d(1,nx,ny,1.,u2,1,md,u2,1,md,table,ntable,
1        work,nwork)
#else
#ifdef sgi
    call cfft2d(-1,nx,ny,u,md,wff1)
    call cfft2d(-1,nx,ny,u2,md,wff1)
#else
    call f2t2d(nx,ny,u,md,u,md,wff1,wff2,work,cpy)
    call f2t2d(nx,ny,u2,md,u2,md,wff1,wff2,work,cpy)
#endif
#endif
        do 105 k = 1, ny
            do 105 jj = 1, nx
                u2(jj,k) = u2(jj,k) * phlin(jj,k)
105         u(jj,k) = u(jj,k) * phlin(jj,k)
110     continue
        endif
c
c output field distributions
c
        energ = 0.
        do 116 k = 1, ny
            do 116 jj = 1, nx
116         energ = energ + real(u(jj,k)*conjg(u(jj,k)))
            energ2 = 0.
            do 316 k = 1, ny
                do 316 jj = 1, nx
316         energ2 = energ2 + real(u2(jj,k)*conjg(u2(jj,k)))
                if (ans2 .eq. 'n') then
                    do 117 k = 1, ntot-7, 8
                        koo3 = koo3 + 1
                        write(7,rec=koo3) (u2(mod(k+k0,nx)+1,(k+k0)/nx+1),k0=-1,6)
117         write(3,rec=koo3) (u(mod(k+k0,nx)+1,(k+k0)/nx+1),k0=-1,6)
                else
#ifdef cray
                    call cfft2d(-1,nx,ny,dntot,u,1,md,u,1,md,table,ntable,
1        work,nwork)
                    call cfft2d(-1,nx,ny,dntot,u2,1,md,u2,1,md,table,ntable,
1        work,nwork)
#else
#ifdef sgi
                    call cfft2d(1,nx,ny,u,md,wff1)
                    call cscal2d(nx,ny,dntot,u,md)

```

```

        call cfft2d(1,nx,ny,u2,md,wff1)
        call cscal2d(nx,ny,dntot,u2,md)
#else
        call f2t2b(nx,ny,u,md,u,md,wff1,wff2,work,cpy)
        call f2t2b(nx,ny,u2,md,u2,md,wff1,wff2,work,cpy)
        do 379 ii = 1, ny
            do 379 ji = 1, nx
                u2(ji,ii) = dntot*u2(ji,ii)
379      u(ji,ii) = dntot*u(ji,ii)
#endif
#endif
c
c  Subtract a reference phase
c
        refph = mod(phzinit+roptnon*float(i),pi2)
        refph2 = mod(phzinit+roptnon*pratio**float(i),pi2)
        write(6,129) refph, refph2
129      format(' max. nonlinear reference phase = ',e14.7)
        call hdf_img2(filnam4,refph,rlv,u,md,nx,ny,i,work,ntot,2)
        call hdf_img2(filnam24,refph2,rlv,u2,md,nx,ny,i,work,ntot,2)
#ifdef cray
        call cfft2d(1,nx,ny,1.,u,1,md,u,1,md,table,ntable,
1          work,nwork)
        call cfft2d(1,nx,ny,1.,u2,1,md,u2,1,md,table,ntable,
1          work,nwork)
#else
#ifdef sgi
        call cfft2d(-1,nx,ny,u,md,wff1)
        call cfft2d(-1,nx,ny,u2,md,wff1)
#else
        call f2t2d(nx,ny,u,md,u,md,wff1,wff2,work,cpy)
        call f2t2d(nx,ny,u2,md,u2,md,wff1,wff2,work,cpy)
#endif
#endif
        endif
120      write(6,*) 4.*zprop, energ*dntot, energ2*dntot
        if (ans2 .eq. 'y') then
            do 217 k = 1, ntot-7, 8
                koo3 = koo3 + 1
                write(7,rec=koo3) (u2(mod(k+k0,nx)+1,(k+k0)/nx+1),k0=-1,6)
217      write(3,rec=koo3) (u(mod(k+k0,nx)+1,(k+k0)/nx+1),k0=-1,6)
            endif
c
c  Propage zdist+z2-zprop distance behind the medium
c
        close (3)
        close (7)
        stop
        end

```

Fig. 8. An example of the input file non2beam.dat in which the data from the user are bold-faced.

```

DATA FOR NON2BEAM.F enter data after this vertical line --> |
Name of the file containing E-field data for beam1           |gaus.d
Name of the file containing E-field data for beam2           |gaus.d

```

Is the above file produced by non2beam.f? (y/n)	n
Frame # to use if the previous answer is y (default 1)	
Initial propagation distance if the previous answer is y (default 0)	
delta n ($n_2 \cdot E_0^2 / n_0$)	1.e-4
Wavelength in meter (default 1.e-6)	
Radial size w0 in meter (default 1.e-4)	
Resolution for x range (<1 for high resolution, default 1)	
Resolution for y range (<1 for high resolution, default 1)	
Name of the file containing FT(field) for beam1 (default ft.d)	
Name of the file containing FT(field) for beam2 (default ft2.d)	
Linear refractive index n0 at the carrier freq (default 1)	
Sign of nonlinearity (-1 or +1) (default -1)	
Max. Length of medium in $z_0 = \pi \cdot n_0 \cdot w_0^2 / \text{wavelength}$	1.
relative error (<1) (default 1.e-3)	
Output images in hdf format? (y/n)	y
If y, give name of the hdf file for beam1 (default ifr.hdf)	
If y, give name of the hdf file for beam2 (default ifr2.hdf)	
No. of frames ipr write to file (default 5)	
Distance of the beam waist in front of the medium (in Z0)	0.
Power of beam1 / Power of beam2	1.
Coupling coef. between the two beams	0.

The explanations for the input file input.dat in Fig. 8 are as follows:

- line 1: Name of the file contains input electric field data for the first beam or circularly polarized component in near field or in far field.
- line 2: Name of the file contains input electric field data for the second beam or circularly polarized component in near field or in far field.
- line 3: Enter 'n' for near field data or 'y' for far field data which is the output format of non2beam.F. This option is for continuous execution of the program.
- line 4: If data for line 3 is 'y', enter the frame number to pick from the input field data which may contain data at different propagation distance.
- line 5: If data for line 3 is 'y', enter the propagation distance corresponding to the frame picked.
- line 6: Maximum change in refractive index due to nonlinearity.
- line 7: Wavelength of the carrier.
- line 8: Scale factor in the transverse dimensions.
- line 9: Resolution in x dimension.
- line 10: Resolution in y dimension.
- line 11: Name of the file contains the far field data for the first beam which may include data for all frames or only the data for the last frame when output in HDF images is specified.
- line 12: Name of the file contains the far field data for the second beam which may include data for all frames or only the data for the last frame when output in HDF images is specified.
- line 13: Linear refractive index.

- line 14: Sign of the nonlinearity.
- line 15: Length of the nonlinear medium.
- line 16: Error control parameter.
- line 17: Enter 'y' for output data in HDF format.
- line 18: If answer in line 17 is 'y', enter name of the file containing the HDF images for the first beam.
- line 19: If answer in line 17 is 'y', enter name of the file containing the HDF images for the second beam.
- line 20: Number of output frames written to the far field data files or HDF images files.
- line 21: Propagation distance in front of the nonlinear medium.
- line 22: Power ratio of the first beam and the second beam.
- line 23: The ratio between self-phase modulation and cross-phase modulation.

4. Presentations / publications under the Grant during 1993-94

- [1] C. T. Law and G. A. Swartzlander, Jr., "Polarized Optical Vortex Solitons: Instabilities and Dynamics in Kerr Nonlinear Media," to appear in *Chaos, Solitons & Fractals* (1994).
- [2] C. T. Law and G. A. Swartzlander, Jr., "Polarized vortex dynamics in Kerr nonlinear material," to be presented in CLEO '94 Anaheim, California (May 8-13, 1994).
- [3] C. T. Law and G. A. Swartzlander, Jr., "Wave guiding by optical vortex solitons," in *OSA Annual Meeting*, OSA Annual Meeting Technical Digest, 1993, vol. 16 (Optical Society of America, Washington, DC, 1993).
- [4] G. A. Swartzlander, Jr. and C. T. Law, "Optical-vortex solitons in Kerr nonlinear media," in *Quantum Electronics and Laser Science Conference*, 1993 OSA Technical Digest Series, Vol. 3 (Optical Society of America, Washington, DC, 1993).